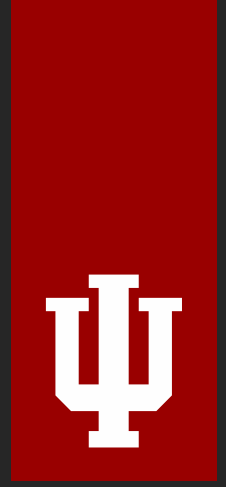


Introduction to R and R Studio

Lilian Golzarri-Arroyo

Biostatistics Consulting Center
School of Public Health - Bloomington

https://go.iu.edu/2bZY

Section 1

Introductions

Introductions

- **Biostatistics Consulting Center**, School of Public Health --- biostats.indiana.edu

  Professional statisticians for health-related research. Free consulting Tu/Th 10-12 @ SSRC.

- **Social Science Research Commons** --- ssrc.indiana.edu
- **Research Analytics**, UITS RT --- https://rt.iu.edu/
- **Indiana Statistical Consulting Center**, --- iscc.indiana.edu
- **Center for Survey Research** --- csr.indiana.edu

Statistical Software

- **R –** syntax code, free & flexible
- **SAS –** syntax code, industry standard
- **SPSS –** easy “point & click”, good for most “off the shelf” analyses
- **STATA –** syntax with “point & click”, political science, economics, sociology
- **JMP –** “point & click”, good mix of stats and graphs – good for exploring data
- **MATLAB –** powerful numerical computing, matrix manipulations

Section 2

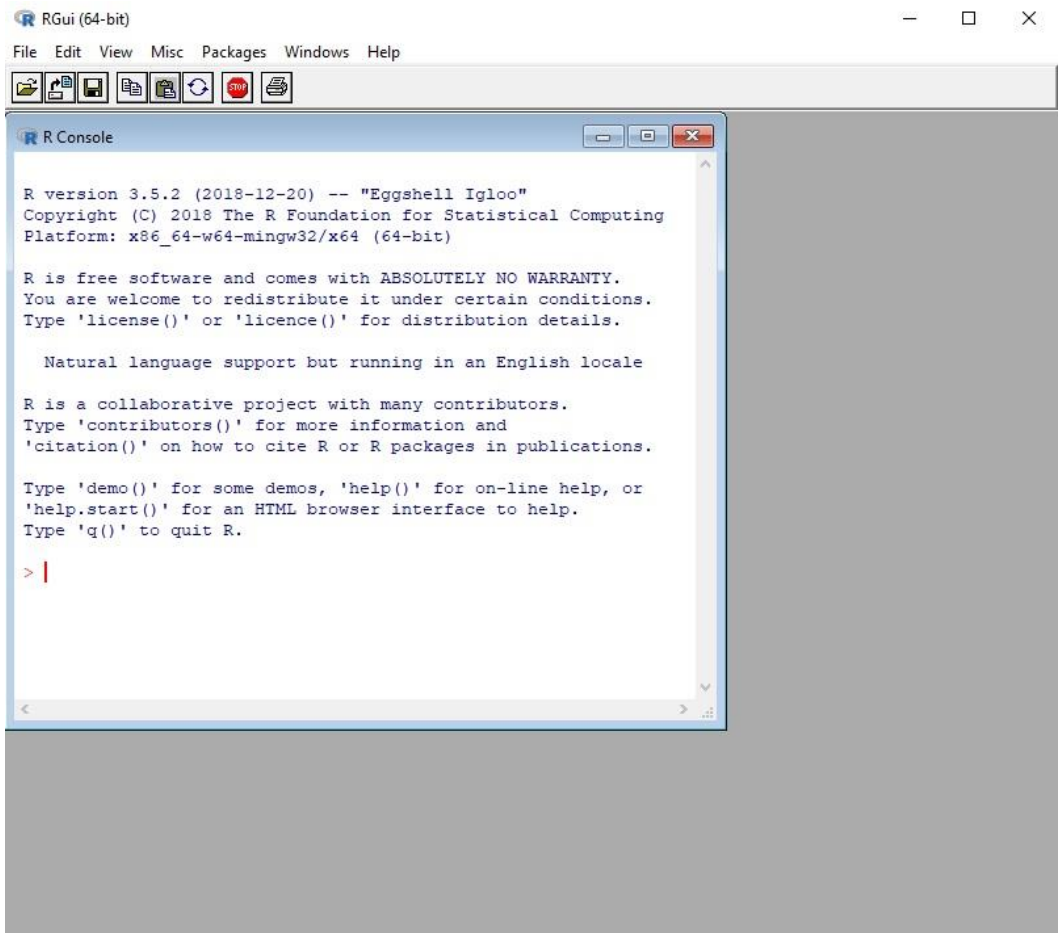
Let’s get started

Getting ready

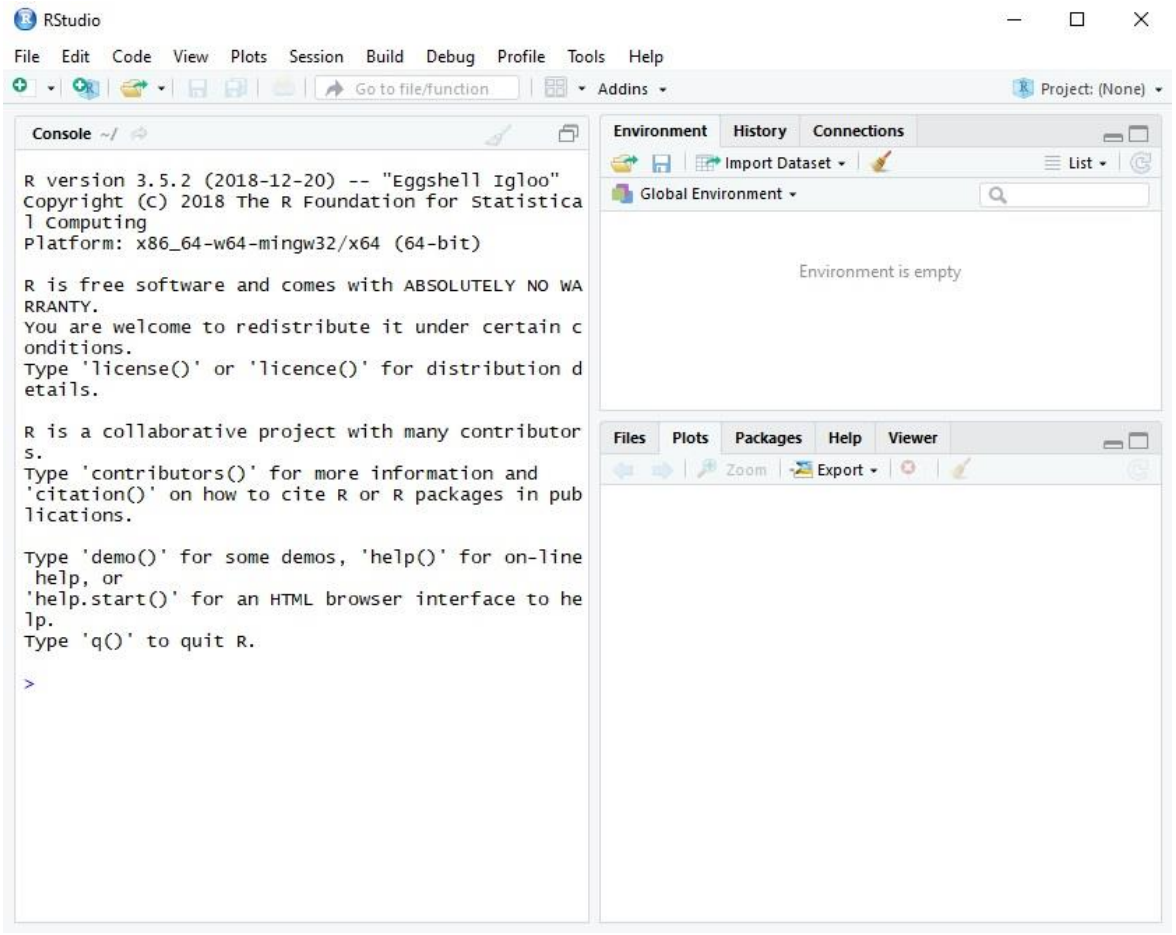
- **To download R:** https://cran.r-project.org/
- **To download R Studio**: https://rstudio.com/
- To download data and slides: https://go.iu.edu/2bZY

- "R is a freely available language and environment for statistical computing and graphics which provides a wide variety of statistical and graphical techniques: linear and nonlinear modelling, statistical tests, time series analysis, classification, clustering, etc." (From https://cran.r-project.org/)

R Console



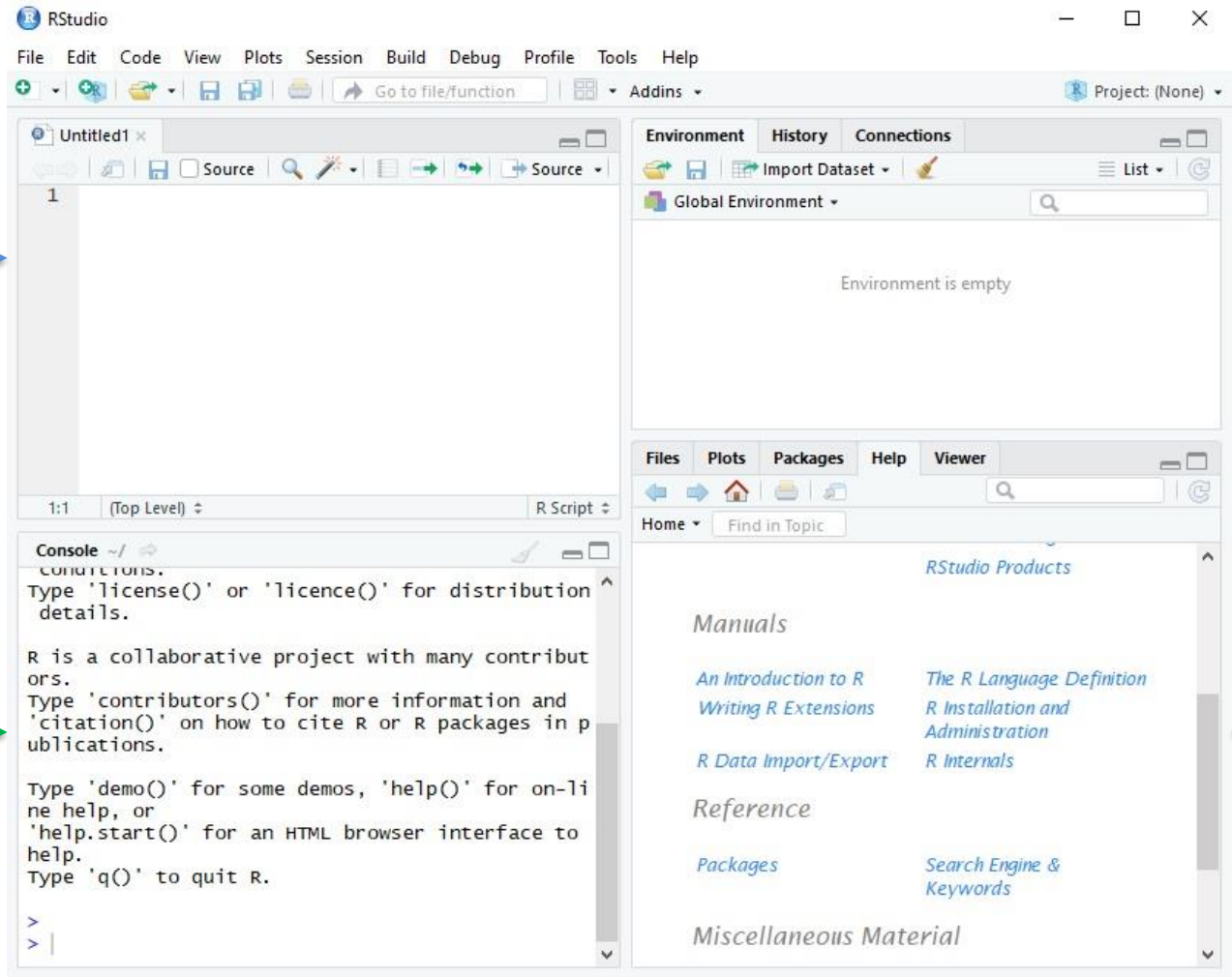
RStudio



R Studio

Editor Window

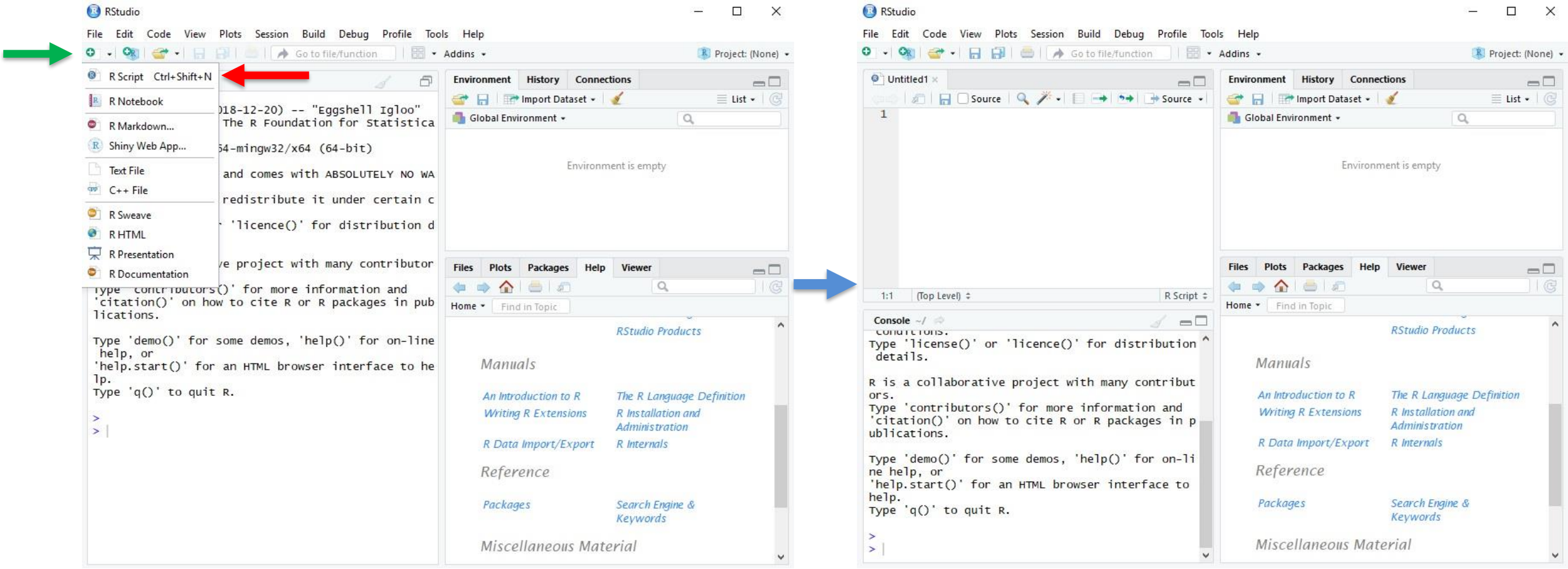
Console Window



Environment Window

Help and Viewer Window

Create New Script



To do list

- Basic operations
- Create variables
- Create vectors
- Functions
- Data type
- Create data frame
- Read data
- Working with data frame
- Summary functions
- Chi-square test
- T-test
- ANOVA
- Correlation
- Linear regression
- ggpplot2

To do list

- Basic operations
- Create variables
- Create vectors
- Functions
- Data type
- Create data frame
- Read data
- Working with data frame
- Summary functions
- Chi-square test
- T-test
- ANOVA
- Correlation
- Linear regression
- ggpplot2

Let’s go to the software

Set directory and read data

Specify with forward slashes or double backslashes
Enclose in single or double quotation marks

Examples
- setwd(“C:/Users/lgolzarr/Desktop”)
- setwd(‘C:\\Users\\lgolzarr\\Desktop’)

Read data in csv
- read.csv(“cats.csv”)
- read.csv(file.choose(),header=TRUE) #If you don’t know the path for the file

Read other data types like txt
- read.table(file, header = TRUE, sep = “”)

sep defines the separator, e.g. “,” or “\t” or “”
header indicates variable names should be read from first row

For other data types there is the **foreign** package that can read SPSS, SAS, STATA files
For Excel files **xlsx** package [read.xlsx("myfile.xlsx", sheetName = "Sheet1")]

Data set “cats”

It is a publicly available data set from library MASS

Description

The heart and body weights of samples of male and female domestic cats used for *digitalis* experiments. The cats were all adult, over 2 kg body weight.

This data frame contains the following columns:

- Sex: Factor with levels "F" and "M".
- Bwt: body weight in kg.
- Hwt: heart weight in g.
- Hwt_NA: heart weight in g with missing values. (I randomly delete some data)
- City: Cities where they live. (I randomly assign cities)

R Help

Helpful commands

- If you know the function name: help() or ?
  - help(cor)
  - ?cor
- If you do not know the function name: help.search() or ??
  - help.search(“t.test”)
  - ??t test

References and more…

- Rstudio cheat sheets: https://rstudio.com/resources/cheatsheets/
- STAT 545 course at the University of British Columbia: https://stat545.com/
- R Programming for Data Science: https://bookdown.org/rdpeng/rprogdatascience/
- Stackoverflow-style sites are great for getting help: https://stackoverflow.com/
- Statistical tools for high-throughput data analysis: http://www.sthda.com/english/
- Or just Google it! https://www.google.com/

Thank you!

```
#-----------------------------------------------#
#     R and RStudio Workshop                    #
#     October 24th, 2019                        #
#                                               #
#     Lilian Golzarri-Arroyo                    #
#     Biostatistics Consulting Center           #
#     School of Public Health - Bloomington     #
#-----------------------------------------------#

#-----Use as calculator-----#
1+1
1+2*3
(1+2)*3
2^3


#-----Use to compare values and get "logical" values
# ( == , <= , >= , > , < , != )-----#
2*2 == 4
2*2 >= 5


#-----Create variables-----#

# Assing a number to a variable
weight_kg <- 60
60 -> weight_kg
weight_kg = 60
weight_kg

# Perform calculations with variables
weight_lb <- weight_kg*2.2
weight_lb

# Reassign variable
weight_kg<- 100
weight_kg


# Remember R is CASE SENSITIVE
weight_KG # We'll get an error since this is not the name of our variable


#-----Create vectors and working with vectors-----#
x <- c(10, 20, 30, 40, 50)

# Calculation with vectors
x+5

y <- c(2,3,4,5,6)
```

```
# Calculation between vectors
x+y

# New vectors using past vectors
z <- c(x,y)
z

# First value of the vector
x[1]

# Fifth value of vector
y[5]

# Add first value of vector x and fifth value of vector y
x[1] + y[5]

# Take values third to fifth from x vector
x[3:5]


#-----String vector-----#

# Define vector of strings, use quotation marks for each element
mysentence <- c("Hello", "World", "Sunshine")

# Take second element
mysentence[2]

# Show first and third element
mysentence[c(1,3)]
mysentence[-2]


#-----Built-in Functions-----#

# Sum all values in vector
?sum    #Ask for help in R
sum(x)

# Take the mean and median for all values
mean(x)
median(x)

# Obtain summary statistics for vector
summary(z)

#Create histogram of value
hist(z)

# Repeat number six, ten times
```

```
rep(6,times=10)
rep(6,10)

# Repeat a vector of numbers
rep(c(1,2,3), 5)
rep(y,5)

# It also work for string vector
rep(c("One", "Two", "Three"), 5)

# Repeat each number 3 times
rep(c(4,5,6), 3) #wrong
rep(c(4,5,6), each=3)
rep(c(4,5,6), each=3, times=2)


#-----Data type-----#

# For numeric vector
class(x)
str(x)

# For string vector
class(mysentence)
str(mysentence)

# Change to character
as.character(y)
y2 <- as.character(y)
class(y2)

# Change to number
as.numeric(y2)
y3 <- as.numeric(y2)
class(y3)


#-----Create data frame-----#

# Create new variables
data.frame(var1=c(1,2,3,4), var2=c(5,6,7,8))

# Save new data set
newdf <- data.frame(var1=c(1,2,3,4), var2=c(5,6,7,8), var3=c("a","b","c","d"))
newdf

# Concatenate existing vectors , they have to be the same length
data.frame(x,y)
```

```
#-----Set directory-----#
setwd("C:/Users/YourPath/YourFolder")


#-----Reading files-----#
cats <- read.csv("cats.csv")
cats2 <- read.csv(file.choose())

# Type of data
class(cats)
str(cats)


#-----Work with data frame-----#

# Get second column from data set
cats[,2]

# Get second row from data set
cats[2,]

# Get rows 98 to 101
cats[98:101,]

# Get second and thrid columns for rows 98 to 10
cats[98:101,2:3]

# Get variable Heart weight (Hwt)
cats[,4]
cats$Hwt


#-----Summary functions-----#

# Mean of variable
mean(cats$Hwt)
sd(cats$Hwt)
summary(cats$Hwt)
hist(cats$Hwt)
summary(cats)

# Mean of variable with missing data
mean(cats$Hwt_NA)

mean(cats$Hwt_NA, na.rm=TRUE)


# Subset data set
females <- subset(cats, cats$Sex == "F")
males <- subset(cats, cats$Sex == "M")
```

```
# Create new variable (Body weight on pounds)
cats$Bwt_lb <- cats$Bwt*2.2

# Recode variable (Above the mean body weight o below the mean)
cats$Bwt_range <- ifelse(cats$Bwt <= mean(cats$Bwt),"Low","High")

cats$Bwt_range2[cats$Bwt <= mean(cats$Bwt)] <- "Low"
cats$Bwt_range2[cats$Bwt > mean(cats$Bwt)] <- "High"


#-----Chi-square test-----# Compare distribution in two categorical variables

# crosstab table
table(cats$Sex,cats$City)

# Chi square test
chisq.test(cats$Sex,cats$City)


#-----T-test-----# Compare mean of two groups

# compare the mean between males and females
t.test(females$Hwt, males$Hwt) #Not assuming equal variance
t.test(females$Hwt, males$Hwt, var.equal=TRUE) #Not assuming equal variance

#t.test(females$Hwt, males$Hwt, paired=TRUE) #To do a paired t-test


# Boxplot to see difference
plot(Hwt~Sex, data=cats)


#-----ANOVA-----# compare 3 or more groups

#Anova test
anova1 <- aov(Bwt ~ City, data=cats)
summary(anova1)

# Boxplot to see the groups
plot(Hwt~City, data=cats)


#-----Correlation-----#

# Pearson correlation
cor(cats$Bwt,cats$Hwt,method = "pearson")

# Spearman correlation
cor(cats$Bwt,cats$Hwt,method = "spearman")
```

```
# Pearson correlation test
cor.test(cats$Bwt, cats$Hwt,method = "pearson")

# Scatter plot to see correlation
plot(cats$Bwt, cats$Hwt)


#-----Linear Regression-----# Look for association between variables

# Linear regression, dependent variable Heart weight and independent variable Body
weight
mod1 <- lm(Hwt ~ Bwt, data = cats)
summary(mod1)

# Scatter plot to see correlation with regression line
plot(cats$Bwt, cats$Hwt)
abline(lm(Hwt ~ Bwt, data = cats))


#-----GGPLOT2!!!!-----#

# Install packages - Only the first time you need them, afterwards they are already
in your session
install.packages("ggplot2")

# Load packages - each time you open R or RStudio
library(ggplot2)

# Boxplot of Heart weight by Sex
p <- ggplot(data=cats, aes(x=Sex, y=Hwt, fill=Sex)) + #Define data for plot
  geom_boxplot()    #Choose plot

p #print plot

p + labs(title = "Boxplot for Heart Weight by Sex", y="Heart Weight")  #add labels
p + theme_classic() #Change to white background
p + scale_fill_brewer(palette="Dark2") #choose from brewer palettes

# Now everything together!
p <- ggplot(data=cats, aes(x=Sex, y=Hwt, fill=Sex)) + #Define data for plot
  geom_boxplot() +   #Choose plot
  labs(title = "Boxplot for Heart Weight by Sex", y="Heart Weight") + #add labels
  theme_classic() + #Change to white background
  scale_fill_brewer(palette="Dark2") #choose from brewer palettes

p #print plot


# Scatter plot for Heart weight and body weight
```

```
q <- ggplot(data=cats, aes(x=Bwt, y=Hwt)) + #Define data for plot
  geom_point(size=2, shape=17, color="red") +   #Choose plot
  labs(title = "Scatter plot for Heart weight and body weight", y="Heart Weight",
x="Body Weight")  #add labels

q #print plot

q + geom_smooth(method='lm',formula=y~x, linetype="dashed") #Add regression line

q <- ggplot(data=cats, aes(x=Bwt, y=Hwt, colour=Sex)) + #Define data for plot
  geom_point() +   #Choose plot
  labs(title = "Scatter plot for Heart weight and body weight", y="Heart Weight",
x="Body Weight")  #add labels
q

q + geom_smooth(method='lm',formula=y~x)
```

ID	Sex	Bwt	Hwt	Hwt_NA	City
1	F	2	7	7	Chicago
2	F	2	7.4		Bloomington
3	F	2	9.5	9.5	Chicago
4	F	2.1	7.2	7.2	Indy
5	F	2.1	7.3	7.3	Bloomington
6	F	2.1	7.6	7.6	Indy
7	F	2.1	8.1	8.1	Indy
8	F	2.1	8.2		Bloomington
9	F	2.1	8.3	8.3	Chicago
10	F	2.1	8.5	8.5	Indy
11	F	2.1	8.7	8.7	Chicago
12	F	2.1	9.8	9.8	Indy
13	F	2.2	7.1	7.1	Indy
14	F	2.2	8.7	8.7	Chicago
15	F	2.2	9.1	9.1	Indy
16	F	2.2	9.7	9.7	Bloomington
17	F	2.2	10.9		Bloomington
18	F	2.2	11	11	Chicago
19	F	2.3	7.3	7.3	Chicago
20	F	2.3	7.9	7.9	Indy
21	F	2.3	8.4	8.4	Bloomington
22	F	2.3	9	9	Indy
23	F	2.3	9	9	Bloomington
24	F	2.3	9.5	9.5	Indy
25	F	2.3	9.6	9.6	Indy
26	F	2.3	9.7	9.7	Bloomington
27	F	2.3	10.1	10.1	Bloomington
28	F	2.3	10.1	10.1	Bloomington
29	F	2.3	10.6	10.6	Chicago
30	F	2.3	11.2	11.2	Indy
31	F	2.4	6.3	6.3	Indy
32	F	2.4	8.7	8.7	Indy
33	F	2.4	8.8	8.8	Indy
34	F	2.4	10.2	10.2	Chicago
35	F	2.5	9		Bloomington
36	F	2.5	10.9	10.9	Bloomington
37	F	2.6	8.7	8.7	Indy
38	F	2.6	10.1	10.1	Chicago
39	F	2.6	10.1	10.1	Chicago
40	F	2.7	8.5	8.5	Bloomington
41	F	2.7	10.2	10.2	Chicago
42	F	2.7	10.8	10.8	Indy
43	F	2.9	9.9	9.9	Bloomington
44	F	2.9	10.1		Indy
45	F	2.9	10.1	10.1	Chicago
46	F	3	10.6	10.6	Indy

ID	Sex	Bwt	Hwt	Hwt_NA	City
47	F	3	13	13	Indy
48	M	2	6.5	6.5	Chicago
49	M	2	6.5	6.5	Chicago
50	M	2.1	10.1	10.1	Bloomington
51	M	2.2	7.2	7.2	Chicago
52	M	2.2	7.6	7.6	Bloomington
53	M	2.2	7.9	7.9	Bloomington
54	M	2.2	8.5	8.5	Indy
55	M	2.2	9.1	9.1	Chicago
56	M	2.2	9.6	9.6	Indy
57	M	2.2	9.6	9.6	Bloomington
58	M	2.2	10.7	10.7	Chicago
59	M	2.3	9.6		Indy
60	M	2.4	7.3	7.3	Chicago
61	M	2.4	7.9	7.9	Bloomington
62	M	2.4	7.9	7.9	Chicago
63	M	2.4	9.1	9.1	Chicago
64	M	2.4	9.3	9.3	Chicago
65	M	2.5	7.9	7.9	Chicago
66	M	2.5	8.6	8.6	Indy
67	M	2.5	8.8	8.8	Indy
68	M	2.5	8.8	8.8	Indy
69	M	2.5	9.3	9.3	Indy
70	M	2.5	11	11	Indy
71	M	2.5	12.7	12.7	Chicago
72	M	2.5	12.7	12.7	Chicago
73	M	2.6	7.7	7.7	Bloomington
74	M	2.6	8.3	8.3	Chicago
75	M	2.6	9.4	9.4	Indy
76	M	2.6	9.4	9.4	Chicago
77	M	2.6	10.5		Bloomington
78	M	2.6	11.5	11.5	Chicago
79	M	2.7	8	8	Bloomington
80	M	2.7	9	9	Chicago
81	M	2.7	9.6	9.6	Bloomington
82	M	2.7	9.6	9.6	Bloomington
83	M	2.7	9.8	9.8	Indy
84	M	2.7	10.4	10.4	Indy
85	M	2.7	11.1	11.1	Indy
86	M	2.7	12	12	Bloomington
87	M	2.7	12.5	12.5	Bloomington
88	M	2.8	9.1	9.1	Indy
89	M	2.8	10	10	Bloomington
90	M	2.8	10.2	10.2	Indy
91	M	2.8	11.4	11.4	Bloomington
92	M	2.8	12	12	Chicago

ID	Sex	Bwt	Hwt	Hwt_NA	City
93	M	2.8	13.3	13.3	Chicago
94	M	2.8	13.5	13.5	Chicago
95	M	2.9	9.4	9.4	Indy
96	M	2.9	10.1	10.1	Chicago
97	M	2.9	10.6	10.6	Bloomington
98	M	2.9	11.3	11.3	Bloomington
99	M	2.9	11.8	11.8	Chicago
100	M	3	10		Bloomington
101	M	3	10.4	10.4	Indy
102	M	3	10.6	10.6	Indy
103	M	3	11.6	11.6	Indy
104	M	3	12.2	12.2	Indy
105	M	3	12.4	12.4	Chicago
106	M	3	12.7	12.7	Indy
107	M	3	13.3	13.3	Bloomington
108	M	3	13.8	13.8	Bloomington
109	M	3.1	9.9	9.9	Indy
110	M	3.1	11.5	11.5	Bloomington
111	M	3.1	12.1	12.1	Bloomington
112	M	3.1	12.5	12.5	Bloomington
113	M	3.1	13	13	Chicago
114	M	3.1	14.3	14.3	Indy
115	M	3.2	11.6	11.6	Bloomington
116	M	3.2	11.9	11.9	Chicago
117	M	3.2	12.3	12.3	Chicago
118	M	3.2	13		Chicago
119	M	3.2	13.5	13.5	Chicago
120	M	3.2	13.6	13.6	Indy
121	M	3.3	11.5	11.5	Indy
122	M	3.3	12	12	Chicago
123	M	3.3	14.1	14.1	Bloomington
124	M	3.3	14.9	14.9	Bloomington
125	M	3.3	15.4	15.4	Chicago
126	M	3.4	11.2	11.2	Chicago
127	M	3.4	12.2		Bloomington
128	M	3.4	12.4	12.4	Bloomington
129	M	3.4	12.8	12.8	Bloomington
130	M	3.4	14.4	14.4	Indy
131	M	3.5	11.7	11.7	Bloomington
132	M	3.5	12.9	12.9	Chicago
133	M	3.5	15.6	15.6	Chicago
134	M	3.5	15.7	15.7	Indy
135	M	3.5	17.2	17.2	Chicago
136	M	3.6	11.8	11.8	Bloomington
137	M	3.6	13.3		Indy
138	M	3.6	14.8	14.8	Indy

ID	Sex	Bwt	Hwt	Hwt_NA	City
139	M	3.6	15	15	Bloomington
140	M	3.7	11	11	Chicago
141	M	3.8	14.8	14.8	Chicago
142	M	3.8	16.8	16.8	Bloomington
143	M	3.9	14.4		Bloomington
144	M	3.9	20.5	20.5	Chicago